

## WHAT IS CLAIMED IS:

5 1. A method for recovering 3D scene structure and camera motion from image data obtained from a multi-image sequence, wherein a reference image of the sequence is taken by a camera at a reference perspective and one or more successive images of the sequence are taken at one or more successive different perspectives by translating and/or rotating the camera, the method comprising the steps of:

(a) determining image data shifts for each successive image with respect to the reference image; the shifts being derived from the camera translation and/or rotation from the reference perspective to the successive different perspectives;

(b) constructing a shift data matrix that incorporates the image data shifts for each image;

(c) calculating a rank-1 factorization from the shift data matrix using SVD, with one of the rank-1 factors being a vector corresponding to the 3D structure and the other rank-1 factor being a vector corresponding to the size of the camera motions;

(d) dividing the successive images into smoothing windows;

(e) recovering the direction of camera motion from the first vector corresponding to the 3D structure by solving a linear equation; and

(f) recovering the 3D structure by solving a linear equation using the recovered camera motion.

2. The method of claim 1, wherein, step (e) includes:  
computing a first projection matrix;

recovering camera rotation vectors from the shift data matrix, and the first projection matrix;

computing a second projection matrix; and

recovering the direction of camera translation using the shift data matrix, the reference image, the second projection matrix and the recovered camera rotation vectors.

3. The method of claim 2, wherein step (f) includes recovering the 3D structure from the shift data matrix, the reference image, the recovered camera rotation vectors and the recovered direction of translation vectors.

4. The method of claim 1, further including preliminary steps of:  
recovering the rotations of the camera between each successive image; and  
warping all images in the sequence toward the reference image, while neglecting the translations.

5. The method of claim 1, wherein step (b) comprises:

computing  $H$  and  $\Delta_{CH}$ , where  $H$  is a  $(N_p - 3) \times N_p$  matrix defined so that  $HH^T$  is the identity matrix and  $H$  annihilates the three vectors  $\Psi_x, \Psi_y, \Psi_z$  where the three vectors are computed from the reference image as

$$\Psi_x \equiv \{\nabla I \cdot \mathbf{r}^{(1)}(p)\}, \Psi_y \equiv \{\nabla I \cdot \mathbf{r}^{(2)}(p)\}, \Psi_z \equiv \{\nabla I \cdot \mathbf{r}^{(3)}(p)\} \text{ where}$$

and  $\Delta$  is a shift data matrix, that gives the difference in intensities between each successive image and the reference image and is a  $(N_I - 1) \times N_p$  matrix with entries  $\Delta I_n^i$ , where  $\Delta I_n^i$  is the change in (smoothed) intensity with respect to the reference image,

6. The method of claim 1, wherein step (c) comprises:  
 computing a rank-1 factorization of  $-\Delta_{CH} \approx \mathbf{M}^{(i)} \mathbf{S}^{(i)r}$  where  $\mathbf{M}^{(i)}, \mathbf{S}^{(i)r}$  are vectors corresponding to the motion and structure respectively.

computing a rank-3 factorization of  $-\Delta_{CH} \approx \sum_{a=1}^3 \mathbf{M}^{(a)} \mathbf{S}^{(a)T}$  where  $\mathbf{M}^{(a)}, \mathbf{S}^{(a)T}$  are

setting  $Z_n^{-1}$  as constant within each window, where  $Z$  is the depth from the

NECI1086



listing each of the pixels so that those in the  $k$ -th smoothing window have sequential indices  $n_k, (n_k+1), \dots, (n_{k+1}-1)$ .

10. The method of claim 2, wherein the step of recovering camera rotation vectors includes solving the following equation

$$P_\Omega(H^T S^{(l)} - \Psi_w) = 0 \text{ for the 3-vector } w.$$

NECI1086

$$P_{\hat{T}}(-\hat{T}_x\{I_x\}-\hat{T}_y\{I_y\}+\hat{T}_z\{p\cdot\nabla I\})=0.$$

$(H^T S^{(i)})_n - [\Psi w]_n = Z_n^{-1} (\hat{T}_z p_n - [\hat{T}]_2) \cdot \nabla I_n$  where  $[\hat{T}]_2$  represents  $\hat{T}_x \hat{T}_y$  the x and y component of the translation direction.

add  
AH

all right